

# switch ~ case 문

```
switch(정숫값){  
    case 정숫값 1:  
        실행할 문장 1;  
        break;  
    case 정숫값 2:  
        실행할 문장 2;  
        break;  
    default:  
        실행할 문장 3;  
        break;  
}
```

```
int main(void) {  
    int n;  
    scanf_s("%d", &n);  
  
    switch (n) {  
        case 1:  
            puts("1");  
            break;  
        case 2:  
            puts("2");  
            break;  
        default:  
            puts("?");  
            break;  
    }  
}
```

```
int main(void) {
    int n, mok;
    char grade;
    puts("성적을 입력해 주세요.");
    scanf_s("%d", &n);
    if (n < 0 || n > 100) {
        puts("입력값이 잘못되었습니다.");
    }
    else {
        mok = n / 10;
        switch (mok) {
            case 10: case 9:
                grade = 'A';
                break;
            case 8:
                grade = 'B';
                break;
            default:
                grade = 'F';
                break;
        }
        printf("입력한 성적은 : %d, 학점은 %c 입니다.", n, grade);
    }
}
```

```
int main(void){
    char op;
    int n1, n2;
    printf("첫번째 수를 입력하세요\n");
    scanf_s("%d", &n1);
    printf("연산자를 입력하세요\n");
    scanf_s(" %c", &op, 1);
    printf("두번째 수를 입력하세요\n");
    scanf_s("%d", &n2);
```

선택 Microsoft Visual Studio 디버그 콘솔

```
첫번째 수를 입력하세요
10
연산자를 입력하세요
*
두번째 수를 입력하세요
5
10 * 5 = 50
```

선택 Microsoft Visual Studio 디버그 콘솔

```
첫번째 수를 입력하세요
10
연산자를 입력하세요
0
} 두번째 수를 입력하세요
2
연산자가 잘못되었습니다.
```

```
#include <stdio.h>
```

```
int main(void){  
    char a;  
    puts("문자를 입력해주세요.");  
    scanf_s("%c", &a, 1);  
    switch (a) {  
        case 'a':  
            printf("A\n");  
            break;  
        case 'b':  
            printf("B\n");  
            break;  
        case 'c':  
            printf("C\n");  
            break;  
    }  
}
```

## switch ~ case 문자

# switch ~ case 문으로 계산코드작성

```
int main(void){  
    int a, b, result; // result: 연산결과 저장용  
    char opt;  
    puts("정수 연산자 정수를 입력하세요.");  
    scanf_s("%d %c %d", &a, &opt, 1, &b);
```

# switch ~ case 문으로 계산코드작성

```
switch(정숫값){  
    case 정숫값 1:  
        실행할 문장 1;  
        break;  
    case 정숫값 2:  
        실행할 문장 2;  
        break;  
    default:  
        실행할 문장 3;  
        break;  
}
```

```
switch (opt) {  
    case '+':  
        result = a + b;  
        break;  
    case '-':  
        result = a - b;  
        break;  
    case '*':  
        result = a * b;  
        break;  
    case '/':  
        result = a / b;  
        break;  
    default:  
        printf("연산자가 잘못되었습니다.\n");  
        flag = 0;  
}  
if (flag) {  
    printf("%d %c %d = %d\n", a, opt, b, result);  
}
```

Microsoft Visual Studio 디버그 콘솔

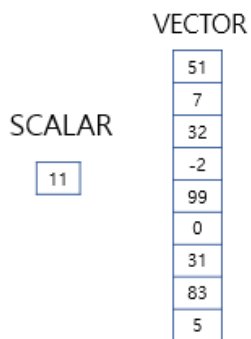
정수 연산자 정수 형태로 입력  
302 \* 459  
302 \* 459 = 138618

Microsoft Visual Studio 디버그 콘솔

정수 연산자 정수 형태로 입력  
10 W 5  
연산자가 잘못되었습니다.

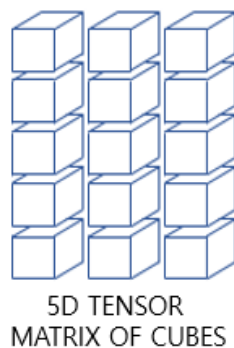
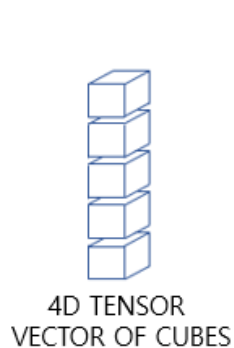
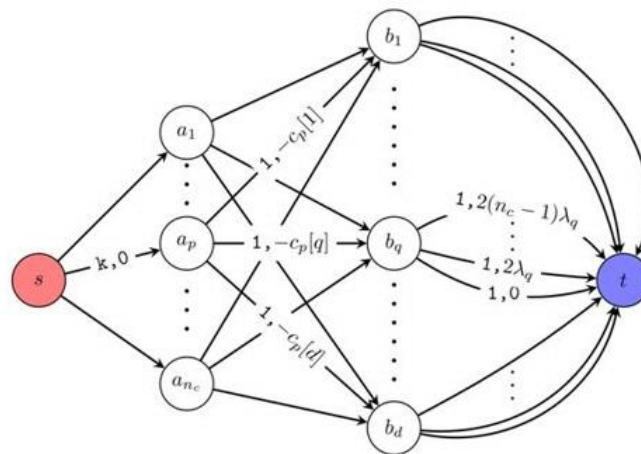
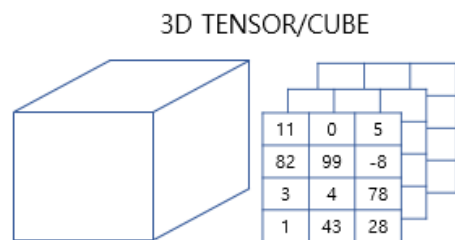
# 반복문

반복해서 처리해야할 모든작업

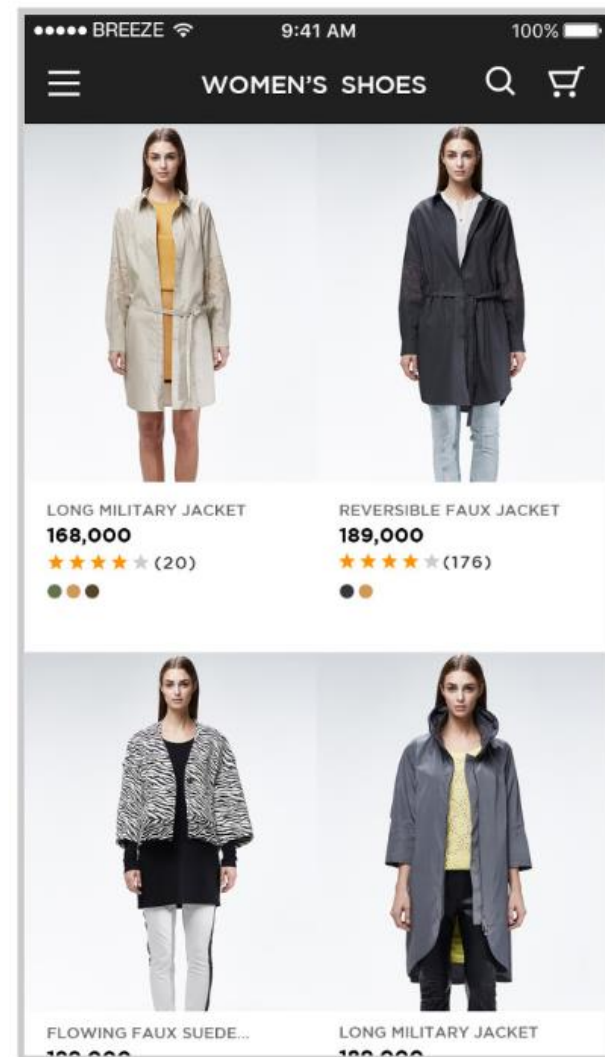


MATRIX

11	0	5
82	99	-8
3	4	78
1	43	28



$$U_1 = M + \sum_{i=1}^n x_i$$



## while 문

```
while(조건식)
{
    반복실행 코드
}

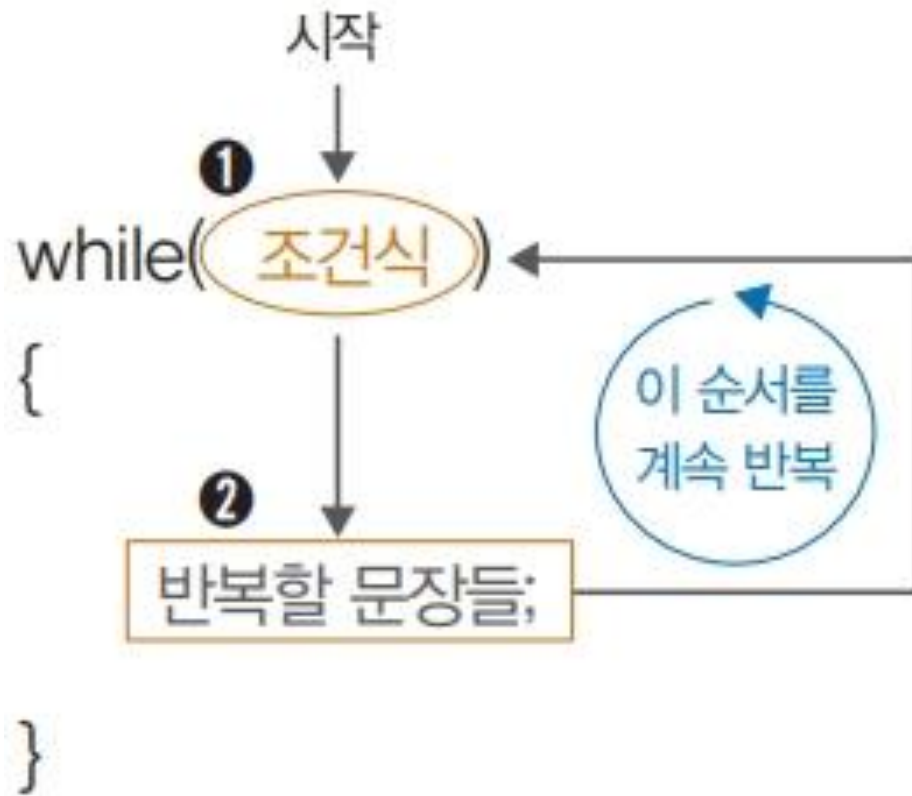
do
{
    반복실행 코드
} while(조건식)
```

## for 문

```
for(초기값 ; 조건식 ; 증감식)
{
    반복실행 코드
}
```



# while 문



```
int main(void){  
    while (1) {  
        printf("hello\n");  
    }  
}
```

Ctrl + c !!!

# while 문

```
int main(void){  
    int n = 1;  
    while (n <= 10) {  
        printf("%d번째 출력\n", n);  
        n = n + 1;  
    }  
}
```

# while 문

```
int main(void){  
    int n = 1;  
    int line;  
    puts("출력할 라인갯수를 입력하세요");  
    scanf_s("%d", &line);  
    while (n <= line) {  
        printf("%d번째 출력\n", n);  
        n += 1;  
    }  
}
```

# while 문

```
int main(void){
    int n = 1;
    int last_n;
    puts("x개의 홀수를 출력합니다. x값을 입력해주세요.");
    scanf_s("%d", &last_n);
    while (n <= last_n) {
        printf("%d번째 홀수 : %d\n", n, (n * 2) - 1);
        n++;
    }
}
```

# 짝수로 바꿔 보세요

```
int main(void){  
    int n = 1;  
    int last_n;  
    puts("x개의 짝수를 출력합니다. x값을 입력해주세요.");  
    scanf_s("%d", &last_n);  
    while (n <= last_n) {  
        printf("%d번째 짝수 : %d\n", n, n * 2);  
        n++;  
    }  
}
```

Microsoft Visual Studio 디버그 콘솔

```
x개의 짝수를 출력합니다. x값을 입력해주세요.  
10  
1번째 짝수 : 2  
2번째 짝수 : 4  
3번째 짝수 : 6  
4번째 짝수 : 8  
5번째 짝수 : 10  
6번째 짝수 : 12  
7번째 짝수 : 14  
8번째 짝수 : 16
```

# X까지의 짝수로 바꿔 보세요

cs 선택 Microsoft Visual Studio 디버그 콘솔

```
x까지의 짝수를 출력합니다. x를 입력하세요.  
15  
2 4 6 8 10 12 14
```

```
int main(void){  
    int n = 1;  
    int last_n;  
    puts("x까지의 짝수를 출력합니다. x를 입력하세요.");  
    scanf_s("%d", &last_n);  
    while (n * 2 <= last_n) {  
        printf("%d ", n * 2);  
        n++;  
    }  
}
```

```

int main(void) {
    int n, d = 1;
    puts("정수를 입력하면 각 자릿수의 값을 출력합니다.");
    scanf_s("%d", &n);

    while (n != 0) {
        printf("%d의 자리수는 %d입니다.\n", d, n % 10);
        d *= 10;
        n /= 10;
    }
}

```

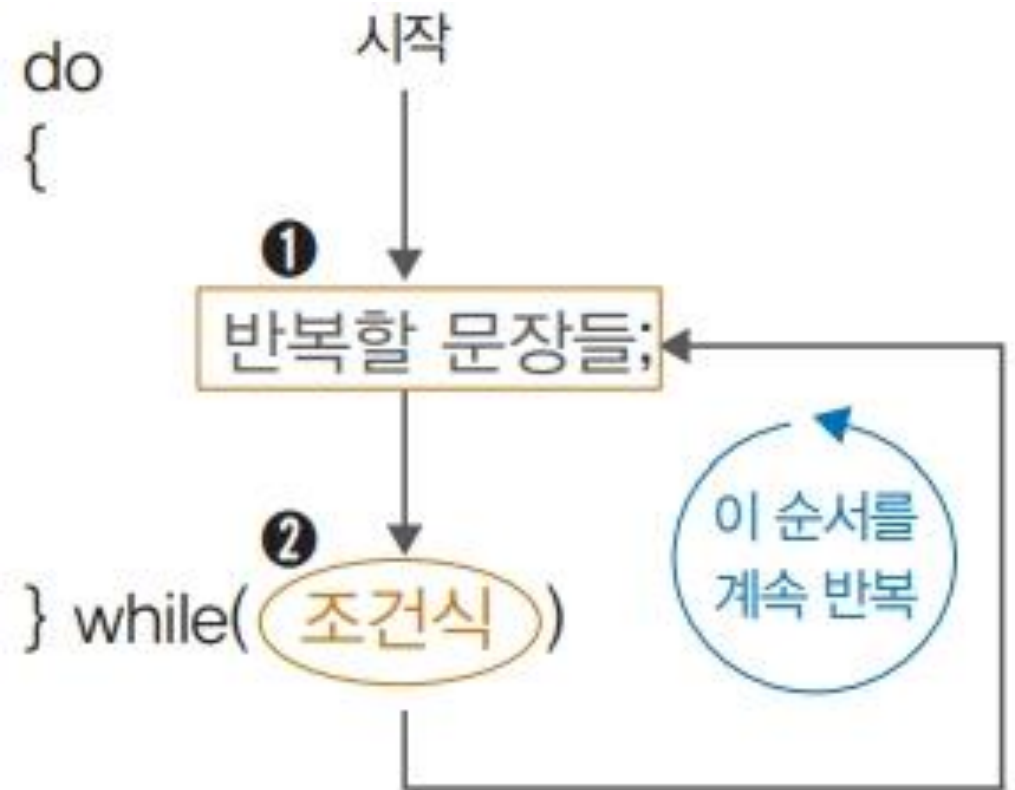
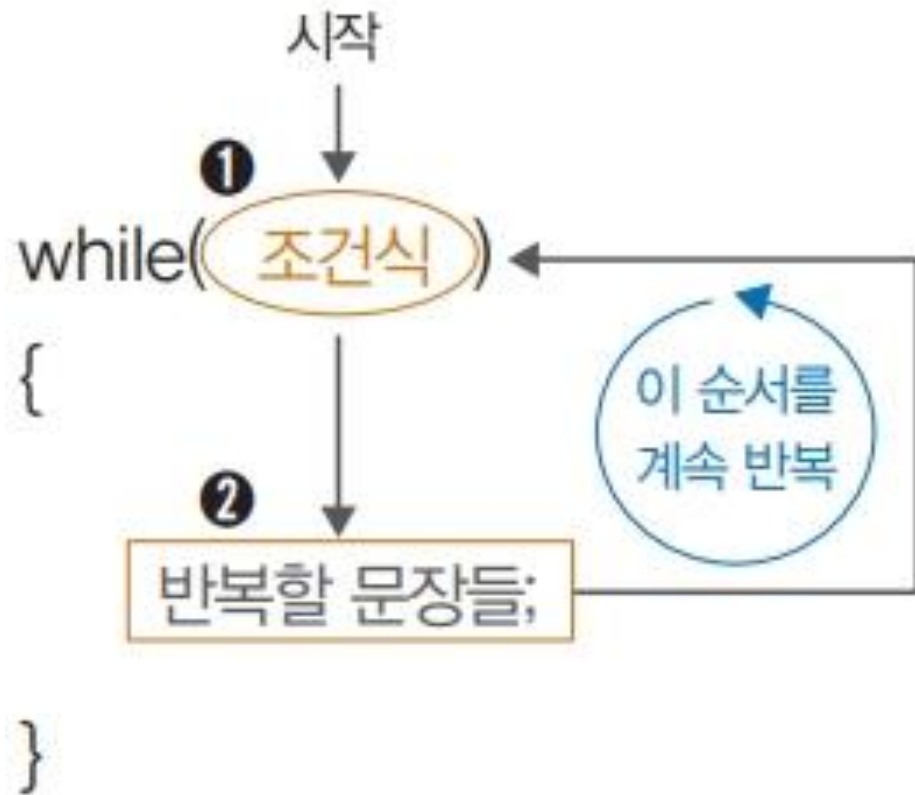
Microsoft Visual Studio 디버그 콘솔

```

정수를 입력하면 각 자릿수의 값을 출력합니다.
1423
1의 자리의 수는 3입니다.
10의 자리의 수는 2입니다.
100의 자리의 수는 4입니다.
1000의 자리의 수는 1입니다.

```

# while과 do while





# do while

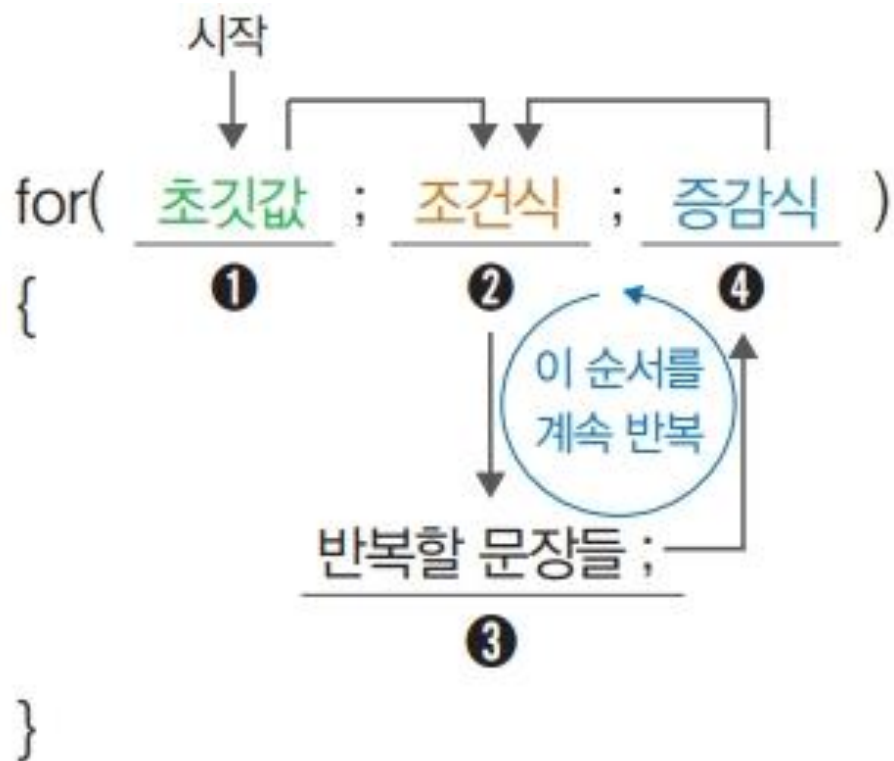
```
int main(void){  
    int n = 200;  
    while (n < 100) {  
        printf("실행문 전에 조건을 검사");  
    }  
}
```

```
int main(void){  
    int n = 200;  
    do {  
        printf("우선 실행 후에 조건을 검사");  
    } while (n < 100);  
}
```

# do while

```
int main(void) {  
    int n, data, result;  
  
    do { // do while 의 예시  
        puts("외부data를 로드합니다.");  
        data = get_data(n); // 외부에서 데이터를 하나씩 로드  
        result = process_data(data); // 정상적으로 처리되면 result가 0  
        n++;  
    } while (result == 0);  
}
```


# for 문



```
int main(void){  
    int i;  
    for ( i = 0; i < 10; i++) {  
        printf( "%d\n", i );  
    }  
}
```

## i : 0부터 사용하는 이유

```
int main(void){ a[0]이기 때문에
    int i;
    int a[5] = { 1, 2, 3, 4, 5 };
    for (i = 0; i < 5; i++) {
        printf("%d\n", a[i]);
    }
}
```



# for문 다양한 형태

for( 초기값 ; 조건식 ; 증감식 )

```
int main(void){  
    for ( ; ; ) {  
        printf("hello!\n");  
    }  
}
```

Ctrl + c !!!

# for문 다양한 형태

for( 초기값 ; 조건식 ; 증감식 )

```
int main(void){  
    int i;  
    i = 1;  
    for ( ; i < 100; ) {  
        printf( "%d\n", i );  
        i *= 5;  
    }  
}
```

# for문 다양한 형태

for( 초기값 ; 조건식 ; 증감식 )

```
int main(void){  
    int i, j;  
    for ( i = 4, j = 8; i * j < 1000; i++, j++) {  
        printf( "%d x %d = %d\n", i, j, i*j );  
    }  
}
```

# for문 중첩

```
int main(void){  
    int i, j;  
  
    for (i = 1; i < 4; i++) {  
        for (j = 1; j < 5; j++) {  
            printf("i=%d, j=%d\n", i, j);  
        }  
    }  
}
```



# 구구단!

```
int main(void){  
    int i, j;  
    for (i = 2; i < 10; i++) {  
        for (j = 1; j < 10; j++) {  
            printf("%d x %d = %d\n", i, j, i * j);  
        }  
    }  
}
```

Microsoft Visual Studio 디버그 콘솔

```
8 x 4 = 32  
8 x 5 = 40  
8 x 6 = 48  
8 x 7 = 56  
8 x 8 = 64  
8 x 9 = 72  
9 x 1 = 9  
9 x 2 = 18  
9 x 3 = 27  
9 x 4 = 36
```

# break 와 continue

반복문( for, while, do~while )

{

⋮

break;

⋮

}



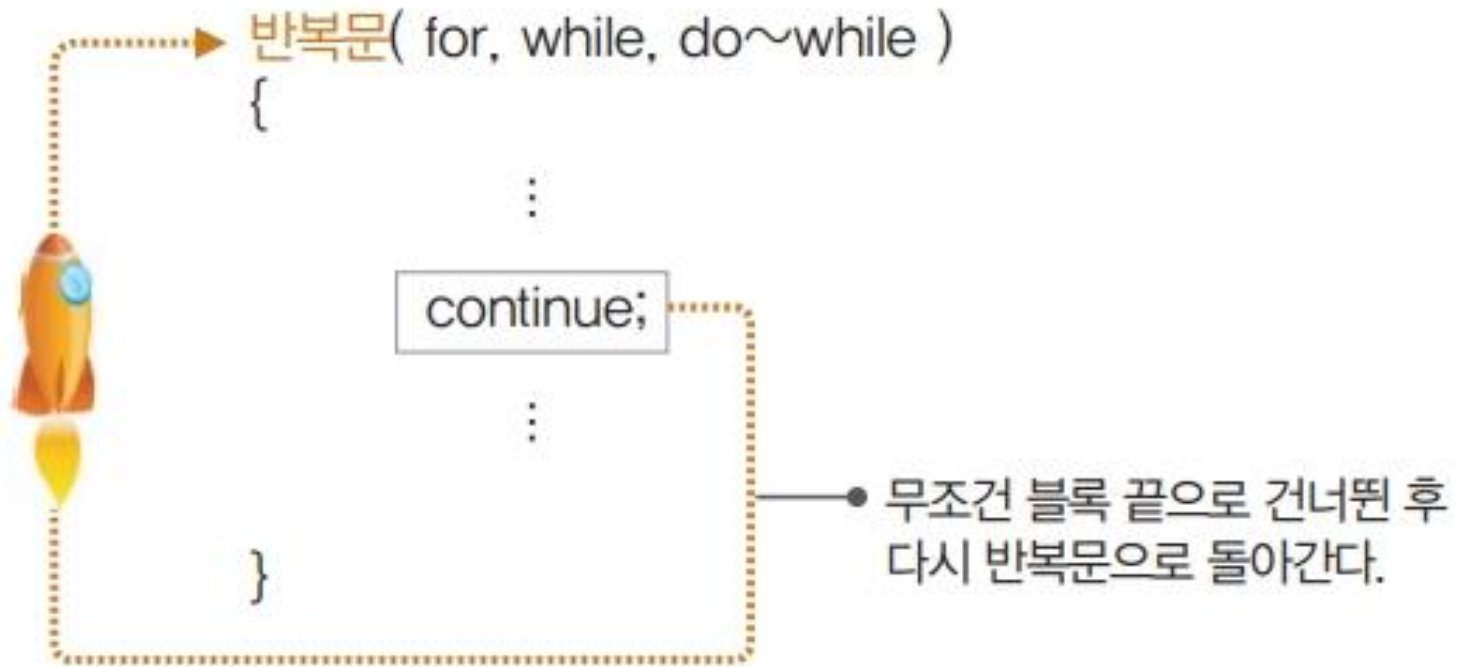
# break

```
int main(void){
    int i, j;
    int last;
    puts("각단의 마지막을 입력하세요");
    scanf_s("%d", &last);
    for (i = 2; i < 10; i++) {
        for (j = 1; j < 10; j++) {
            printf("%d x %d = %d\n", i, j, i * j);
            if (j == last) {
                break;
            }
        }
    }
}
```

# break

```
int main(void){
    int i, j;
    int last;
    puts("마지막 단을 입력하세요");
    scanf_s("%d", &last);
    for (i = 2; i < 10; i++) {
        for (j = 1; j < 10; j++) {
            printf("%d x %d = %d\n", i, j, i * j);
        }
        if (i == last) {
            break;
        }
    }
}
```

# break 와 continue



# continue

```
int main(void){
    int i, j;
    int pass;
    puts("각단의 건너뛰 값을 입력하세요");
    scanf_s("%d", &pass);
    for (i = 2; i < 10; i++) {
        for (j = 1; j < 10; j++) {
            if (j == pass) {
                continue;
            }
            printf("%d x %d = %d\n", i, j, i * j);
        }
    }
}
```

# continue

```
int main(void){
    int i, j;
    int pass;
    puts("건너뛸 단을 입력하세요");
    scanf_s("%d", &pass);
    for (i = 2; i < 10; i++) {
        if (i == pass) {
            continue;
        }
        for (j = 1; j < 10; j++) {
            printf("%d x %d = %d\n", i, j, i * j);
        }
    }
}
```

# 결과값이 짝수인 구구단만 출력

```
int main(void){  
    int i, j;  
  
    for (i = 2; i < 10; i++) {  
        for (j = 1; j < 10; j++) {  
            if ((i * j) % 2 == 1) {  
                continue;  
            }  
            printf("%d x %d = %d\n", i, j, i * j);  
        }  
    }  
}
```

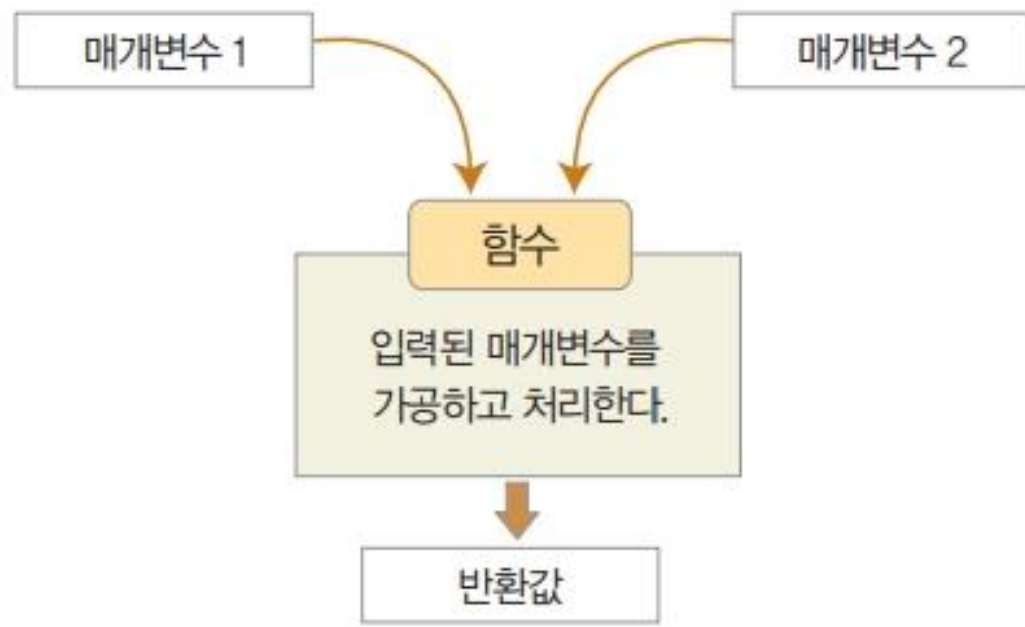
Microsoft Visual Studio 디버그 콘솔

```
6 x 8 = 48  
6 x 9 = 54  
7 x 2 = 14  
7 x 4 = 28  
7 x 6 = 42  
7 x 8 = 56  
8 x 1 = 8  
8 x 2 = 16  
8 x 3 = 24  
8 x 4 = 32
```



# 함수

```
int main(void)
{
    // 실행할 내용
    return 0;
}
```



# 함수 정의

리턴 타입 → 함수 이름 매개 변수

```
int calc(int a, int b, char opt) {  
    int result = 0;  
    switch (opt) {  
        case '+':  
            result = a + b;  
            break;  
        case '-':  
            ...  
    }  
    return result;  
}
```

리턴 값 →

# 함수 선언과 호출

```
#include <stdio.h>
int calc(int, int, char); // calc 함수 선언

int main(void) {
    int a, b, result;
    char opt;
    puts("연산자를 입력하세요(+ - * /, e:종료)");
    scanf_s("%c", &opt, 1);
    puts("계산할 두 수를 입력하세요");
    scanf_s("%d %d", &a, &b);
    result = calc(a, b, opt);
}
```

```
#include <stdio.h>
int calc(int, int, char); // calc 함수 선언
```

```
int main(void){
    int a, b, result;
    char opt;
    do {
        puts("연산자를 입력하세요(+ - * /, e:종료)");
        scanf("%c", &opt);
    } while (opt != 'e');
    puts("종료합니다.");
}
```

C:\Users\origener\Desktop\sbs\_c\c언어\_기초\Debug\Project1.exe

```
연산자를 입력하세요(+ - * /, e:종료)
+
계산할 두 수를 입력하세요
4 5
결과 값: 9
연산자를 입력하세요(+ - * /, e:종료)
```

```
        rewind(stdin);
    }
} while (opt != 'e');
}
```